AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS

1. – 19.   (canceled)

20.   (new) A method of distributed processing, comprising:

providing a processor;

providing a memory having a plurality of memory segments capable of storing either program code or data;

providing a storage location for capable of storing semaphore values each of which are associated with one of said memory segments and operable to indicate whether said associated memory segment contains program code or data that is available for use;

providing a first program and a second program each operable to access the semaphore values;

accessing a first semaphore value by said first program;

determining if the program code or data in the memory segment associated with the first semaphore value is available for use based upon the first semifore value; and

using the first program to implement the code or data stored in the memory segment associated with the first semaphore value.

21.   (new) A method as recited in claim 20, comprising:

altering the first semaphore value by the first program so as to indicate that the memory segment associated with the first semaphore value is available for having program code or data stored therein.

22.   (new) A method as recited in claim 21, comprising:

accessing the altered first semaphore value by the second program;

determining if the memory segment associated with the altered first semaphore value is available to have program code or data stored therein by the second program; and

using the second program to store code or data in the memory segment associated with the semaphore value when the associated memory segment is available.

23.   (new) A method as recited in claim 22, comprising:

completing the storing of the program code or data in the memory segment associated with the semaphore value by the second program; and

altering the semaphore value by the second program to indicate that the program code or data in the memory segment associated with the first semaphore value is available for use by the first program.

24.   (new) A method as recited in claim 1, wherein the memory is a local memory of the processor and wherein the memory segments are logical memory segments implemented by the processor based upon a computer program to be executed by the processor.

25.   (new) A method as recited in claim 20 wherein the storage location is a register, or portions thereof, of the processor or a scalar accessible to the processor.

26.   (new) A method as recited in claim 20, wherein the first program is a routine that is located in a reserved portion of the local memory such that the first program can not be written over with new code or data, and wherein the first program is operable to access the code or data stored in the local memory of the processor as well as implement that accessed code or data.

27.   (new) A method as recited in claim 26, wherein when the program that is operable to implement a portion of a Fast Fourier Transform (FFT) program is stored in the local memory, the first program is operable to access the local memory and begin implementing the FFT code.

28.   (new) A method as recited in claim 27, wherein the second program is operable to load new blocks of code or data that are used by the first program.

29.   (new) A method as recited in claim 28, wherein the second program loads code or data from an external memory to the local memory based upon the first semaphore value.

30.   (new) A method as recited in claim 20, wherein the determining further comprises:

comparing the first semaphore value with a lookup list of pre-determined semaphore values.

31.   (new) Computer program product for distributed processing, comprising:

computer code for providing a processor;

computer code for providing a memory having a plurality of memory segments capable of storing either program code or data;

computer code for providing a storage location for capable of storing semaphore values each of which are associated with one of said memory segments and operable to indicate whether said associated memory segment contains program code or data that is available for use;

computer code for providing a first program and a second program each operable to access the semaphore values;

computer code for accessing a first semaphore value by said first program;

computer code for determining if the program code or data in the memory segment associated with the first semaphore value is available for use based upon the first semifore value;

computer code for using the first program to implement the code or data stored in the memory segment associated with the first semaphore value; and

computer readable medium for storing the computer code.

32. (new) Computer program product as recited in claim 31, comprising:

computer code for altering the first semaphore value by the first program so as to indicate that the memory segment associated with the first semaphore value is available for having program code or data stored therein.

33. (new) Computer program product as recited in claim 32, comprising:

computer code for accessing the altered first semaphore value by the second program;

computer code for determining if the memory segment associated with the altered first semaphore value is available to have program code or data stored therein by the second program; and

computer code for using the second program to store code or data in the memory segment associated with the semaphore value when the associated memory segment is available.

34. (new) Computer program product as recited in claim 33, comprising:

computer code for completing the storing of the program code or data in the memory segment associated with the semaphore value by the second program; and

computer code for altering the semaphore value by the second program to indicate that the program code or data in the memory segment associated with the first semaphore value is available for use by the first program.

35. (new) A method as recited in claim 31, wherein the memory is a local memory of the processor and wherein the memory segments are logical memory segments implemented by the processor based upon a computer program to be executed by the processor.

36. (new) Computer program product as recited in claim 31 wherein the storage location is a register, or portions thereof, of the processor or a scalar accessible to the processor.

37.   (new) Computer program product as recited in claim 31, wherein the first program is a routine that is located in a reserved portion of the local memory such that the first program can not be written over with new code or data, and wherein the first program is operable to access the code or data stored in the local memory of the processor as well as implement that accessed code or data.

38.   (new) Computer program product as recited in claim 37, wherein when the program that is operable to implement a portion of a Fast Fourier Transform (FFT) program is stored in the local memory, the first program is operable to access the local memory and begin implementing the FFT code.

39.   (new) Computer program product as recited in claim 38, wherein the second program is operable to load new blocks of code or data that are used by the first program.

40.   (new) Computer program product as recited in claim 39, wherein the second program loads code or data from an external memory to the local memory based upon the first semaphore value.

41.   (new) Computer program product as recited in claim 31, wherein the determining further comprises:
       computer code for comparing the first semaphore value with a lookup list of pre-determined semaphore values.